

Desarrollo de material docente que integre la implementación del método matricial en  
Python y la validación mediante modelos en OpenSees

Yadith Tatiana Ruiz Mosquera

Trabajo de Grado para Optar el Título de Ingeniera Civil

Director

María Alejandra Oliveros Caicedo

Magister en Ingeniería Civil

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería Civil

Bucaramanga

2026

### **Dedicatoria**

Este proyecto lleva mi nombre, pero no es solo mío. Representa el esfuerzo compartido con quienes me apoyaron a lo largo de este proceso, de voces que susurraron "sigue" cuando el camino se nublaba y de corazones que se hicieron refugio cuando la duda acechaba. Lo dedico, primero y con todo lo que soy, a mi familia. A ustedes, que son el suelo firme bajo mis pies y el horizonte al que miro. Este logro es, en realidad, un espejo donde se refleja cada uno de sus sacrificios, su fe inquebrantable y su amor. Por ustedes, y para ustedes.

### **Agradecimientos**

Al cerrar este capítulo de mi vida, miro atrás y no veo un camino solitario, sino una senda iluminada por rostros que me acompañaron, enseñaron y abrazaron en cada paso. Es por eso que quiero expresar mi gratitud más profunda:

A Dios, por ser la brújula en mi incertidumbre y la paz en mis momentos de caos, por darme la fuerza cuando creía que no había más, y por poner en mi camino a las personas que hoy nombro.

A mis padres y hermanos, mi fortaleza y mi refugio. Gracias por sus sacrificios silenciosos, por sus palabras de aliento en las madrugadas de estudio, por creer en mí incluso cuando yo dudaba. Este título es tan suyo como mío.

A mi directora, María Alejandra, más que un guía académico, fue una presencia humana que supo acompañarme con paciencia cuando el camino se complicaba y con exigencia cuando necesitaba crecer. Cada corrección, cada consejo y cada conversación llevaron la huella de alguien que no solo orienta, sino que genuinamente se compromete con el éxito de sus estudiantes. Le debo no solo este título, sino una forma de entender la dedicación y la responsabilidad profesional.

A mis amigos, aquellos que se convirtieron en familia a lo largo de esta travesía. Por las risas que aliviaron el estrés y por escuchar mis preocupaciones una y otra vez. Con ustedes, la carga se hizo más liviana.

Más allá del título, me llevo el rostro de todos ustedes, tejido en la memoria de estos años. Por eso, al entregar este trabajo, no digo adiós, sino gracias, porque en cada paso que dé en el futuro, llevaré conmigo el eco de su apoyo, su sabiduría y su confianza.

**Tabla de Contenido**

	<b>Pag.</b>
Introducción .....	10
1. Objetivos .....	13
1.1. Objetivo General .....	13
1.2. Objetivos específicos .....	13
2. Marco Pedagógico .....	14
2.1. Método Matricial de Rigidez .....	14
2.2. Lineamientos Curriculares para la Formación en Ingeniería Civil .....	15
2.3. Estrategias de Enseñanza con Herramientas Computacionales .....	15
2.4. Metodologías.....	16
2.5. Evaluación del Aprendizaje .....	17
3. Diseño Metodológico del Proyecto de Aula .....	19
3.1. Contenidos Desarrollados .....	19
3.2. Material Didáctico .....	26
3.3. Estrategias de Evaluación .....	28
4. Desarrollo de la Experiencia Docente.....	30
4.1. Actividades Realizadas .....	30
4.2. Interacción con Estudiantes .....	31
4.3. Evaluación del Proceso .....	32
4.4. Evidencias de Aprendizaje.....	37
5. Reflexión Pedagógica .....	39
Referencias Bibliográficas .....	41
Apéndices.....	45

**Lista de Figuras**

<b>Figura 1</b>	Casos de estudio definidos para armaduras.....	21
<b>Figura 2</b>	Casos de estudio definidos para pórticos .....	22
<b>Figura 3</b>	Casos de estudio definidos para sistemas mixtos.....	23
<b>Figura 4</b>	Desarrollo de sesiones presenciales de la prueba piloto.....	38

## **Lista de Apéndices**

**Apéndice A:** Resultados de la Encuesta Diagnóstica

**Apéndice B:** Resultados de la Validación Computacional de los Casos de Estudio

**Apéndice C:** Guía de Instalación de Anaconda y Configuración del Entorno Virtual

**Apéndice D:** Guías Explicativas

**Apéndice E:** Diapositivas

**Apéndice F:** Videos Tutoriales

**Apéndice G:** Aula Virtual de Aprendizaje Moodle

**Apéndice H:** Encuesta de Inscripción a la Prueba Piloto

**Apéndice I:** Encuesta de Seguimiento Semanal

**Apéndice J:** Encuesta de Evaluación Final de la Prueba Piloto

**Apéndice K:** Consentimiento informado

## Resumen

**Título:** Desarrollo de material docente que integre la implementación del método matricial en Python y la validación mediante modelos en OpenSees \*

**Autor:** Yadith Tatiana Ruiz Mosquera\*\*

**Palabras Clave:** Análisis estructural, método matricial, Python, OpenSees, material docente.

### Descripción:

Este proyecto de grado tuvo como objetivo desarrollar material docente complementario para la enseñanza del análisis estructural, mediante la implementación del método matricial de rigidez en Python y su validación a través del framework de simulación OpenSees. La propuesta surgió como respuesta a la necesidad de integrar las herramientas computacionales empleadas en la asignatura de Análisis Estructural del programa de Ingeniería Civil de la Universidad Industrial de Santander (Excel®, MATLAB®, SAP2000® y OpenSees), con el fin de ofrecer un flujo de trabajo unificado que permitiera al estudiante transitar desde el desarrollo algorítmico propio hasta la verificación computacional. A través de una metodología estructurada en cinco fases, se analizaron nueve casos de estudio bidimensionales clasificados en tres tipologías estructurales: armaduras (elementos biarticulados que transmiten únicamente esfuerzo axial), pórticos (elementos con tres grados de libertad por nodo, capaces de transmitir axial, cortante y momento) y sistemas mixtos (combinación de ambos tipos en un mismo modelo), con tres niveles de complejidad creciente cada una. Para cada caso, se desarrollaron códigos modulares en Python basados en el método matricial, modelos equivalentes en OpenSees para validación cruzada, y recursos didácticos que incluyeron guías explicativas, videos tutoriales y presentaciones. Los resultados comprenden algoritmos documentados y verificados frente a SAP2000® con un error máximo obtenido del 2.32%, dentro del margen de aceptación del 5% definido en el alcance del proyecto, así como una evaluación preliminar mediante una prueba piloto con estudiantes. El trabajo busca fortalecer el proceso de enseñanza-aprendizaje, dotando a los futuros ingenieros civiles de competencias en programación aplicada al análisis estructural.

---

\* Práctica Docente

\*\* Facultad de Fisicomecánicas. Escuela de Ingeniería Civil. Director: María Alejandra Oliveros Caicedo. Magister en Ingeniería civil.

### Abstract

**Título:** Development of teaching materials integrating the implementation of the matrix method in Python and validation through OpenSees models \*

**Autor:** Yadith Tatiana Ruiz Mosquera \*\*

**Palabras Clave:** Structural analysis, matrix method, Python, OpenSees, teaching materials.

### Description:

This undergraduate project aimed to develop supplementary instructional materials for structural analysis teaching by implementing the matrix stiffness method in Python and validating it through the OpenSees simulation framework. The proposal arose in response to the need to integrate the computational tools used in the Structural Analysis course of the Civil Engineering program at the Universidad Industrial de Santander (Excel®, MATLAB®, SAP2000®, and OpenSees), in order to offer a unified workflow allowing students to transition from their own algorithmic development to computational verification. Through a methodology structured in five phases, nine two-dimensional case studies were analyzed, classified into three structural typologies: trusses (bi-articulated elements that transmit only axial forces), frames (elements with three degrees of freedom per node, capable of transmitting axial force, shear, and bending moment), and mixed systems (a combination of both types within a single model), each with three levels of increasing complexity. For each case, modular Python codes based on the matrix method were developed, along with equivalent OpenSees models for cross-validation and didactic resources including explanatory guides, video tutorials, and presentations. The results comprise documented algorithms verified against SAP2000® with a maximum obtained error of 2.32%, within the 5% acceptance threshold defined for the project, as well as a preliminary evaluation through a pilot test with students. This work seeks to strengthen the teaching-learning process by equipping future civil engineers with programming skills applied to structural analysis.

---

\* Teaching Practice

\*\* Department of Physical-Mechanical Engineering. School of Civil Engineering. Advisor: María Alejandra Oliveros Caicedo. M.Sc. Civil Engineering.

### **Reconocimiento de uso de inteligencia artificial**

En el desarrollo del presente documento se hizo uso de herramientas de inteligencia artificial generativa como apoyo en distintas etapas del proceso de escritura académica. Específicamente, se utilizaron los modelos Claude (Anthropic), ChatGPT (OpenAI) y DeepSeek para los siguientes propósitos: estructuración y organización del documento conforme a los lineamientos establecidos, redacción y corrección de texto a partir de ideas, datos y argumentos provistos por el autor.

En todos los casos, el uso de estas herramientas se limitó a funciones de asistencia en la escritura. El contenido conceptual, los datos, los análisis, las decisiones pedagógicas y las reflexiones presentadas a lo largo del documento son de autoría propia. La responsabilidad sobre la veracidad, coherencia y pertinencia del contenido recae enteramente en el autor.

## Introducción

La asignatura de Análisis Estructural en el programa de Ingeniería Civil de la Universidad Industrial de Santander (UIS) se distingue por integrar un componente teórico y uno práctico, consolidando un enfoque formativo en el área de estructuras. Dentro de sus contenidos, el análisis matricial de rigidez representa un eje fundamental, dado que constituye la base de los métodos modernos de cálculo estructural y permite abordar, con rigor y sistematicidad, problemas de creciente complejidad. Sin embargo, la amplitud y profundidad de los temas asociados al método exceden con frecuencia el tiempo disponible en el aula, lo que dificulta una apropiación completa de los conceptos y procedimientos por parte de los estudiantes.

En el desarrollo de esta asignatura se han implementado diferentes herramientas computacionales de acuerdo con la naturaleza de cada componente: en el laboratorio se hace uso de SAP2000®, lo que permite al estudiante familiarizarse con un software de amplia aplicación profesional; mientras que en la parte teórica se han empleado Microsoft Excel®, MATLAB® y Open System for Earthquake Engineering Simulation (OpenSees) (McKenna et al., 2025), con el propósito de facilitar la comprensión de los fundamentos matemáticos y conceptuales. No obstante, este ecosistema de herramientas opera de manera fragmentada: Excel®, si bien útil para automatizar cálculos básicos, presenta limitaciones en cuanto a escalabilidad al enfrentar problemas con un mayor número de elementos, diversos tipos de cargas o condiciones de contorno más complejas. Por su parte, OpenSees, al funcionar en un entorno aislado, no permite al estudiante experimentar un flujo de trabajo integrado que combine el desarrollo propio del algoritmo con su verificación computacional. Esta desconexión constituyó una oportunidad de mejora en el proceso de enseñanza-aprendizaje, con el fin de fortalecer la formación de competencias para la resolución autónoma de problemas.

Para comprender mejor el contexto y las percepciones de los potenciales beneficiarios de este proyecto, se aplicó una encuesta diagnóstica a 55 estudiantes de Análisis Estructural (véase Apéndice A). Los resultados (Tabla A2) revelan que, si bien el 74.5% de los encuestados ha utilizado MATLAB®, el 63.6% emplea herramientas de programación "rara vez" y un 21.8% "nunca" (Tabla A6). Esto sugiere que, aunque los estudiantes tienen exposición a la programación, esta no constituye una práctica consolidada en su rutina académica. Asimismo, un hallazgo clave es que el 70.9% de los estudiantes considera que aprender a programar, particularmente en Python, podría mejorar su rendimiento en Análisis Estructural, y ninguno descartó esta posibilidad (Tabla A5), lo que revela un interés latente por afianzar esta habilidad. En cuanto a estrategias didácticas, la mayoría señaló que los tutoriales en video (67.3%) y los ejercicios prácticos (27.3%) son los recursos más efectivos (Tabla A8), preferencia que orientó el diseño del material desarrollado en el presente proyecto.

Ante este contexto, el presente proyecto desarrolló material docente complementario orientado específicamente al método matricial de rigidez, mediante la incorporación del lenguaje de programación Python en la resolución de ejercicios con un mayor nivel de complejidad progresiva. Python, por su condición de código abierto, su versatilidad, su amplio rango de aplicaciones y su curva de aprendizaje suave (Chimarro-Amaguaña et al., 2023), constituye una herramienta que no solo facilita la construcción de algoritmos, sino que también puede potenciar la autonomía y capacidad de innovación de los estudiantes. Adicionalmente, se aprovechó la integración nativa de Python con OpenSees a través del módulo *openseespy* con el fin de que el estudiante pudiera programar su algoritmo y validarlo en el mismo entorno, buscando ofrecer una alternativa a la visión fragmentada que caracterizaba el uso previo de las herramientas.

Los resultados obtenidos a partir de la implementación del análisis matricial en Python fueron validados mediante OpenSees y contrastados con modelos en SAP2000®, con el fin de garantizar la confiabilidad de los procedimientos. La inclusión de Python no pretendió sustituir los recursos existentes en la asignatura, sino enriquecer el proceso académico y aportar al perfil del futuro ingeniero civil, dotándolo de competencias en programación aplicadas al análisis estructural, estratégicas en un entorno profesional donde la innovación, la adaptación tecnológica y la capacidad de resolver problemas de manera autónoma constituyen atributos altamente valorados.

## 1. Objetivos

### 1.1. Objetivo General

Desarrollar material docente para la enseñanza del análisis estructural, integrando la implementación del método matricial en Python y su validación mediante modelos en OpenSees.

### 1.2. Objetivos específicos

Construir códigos de programación en Python basados en la teoría del método matricial para la resolución de problemas básicos y avanzados de Análisis Estructural, e implementar las librerías de OpenSees en dicho lenguaje con el propósito de validar los resultados obtenidos.

Elaborar recursos de divulgación académica (presentaciones, guías explicativas y videos) que expliquen e interpreten los códigos desarrollados en ejercicios de distintos niveles de complejidad, con el fin de facilitar la socialización y el aprendizaje progresivo del análisis estructural.

Evaluar la utilidad y pertinencia del material docente en el fortalecimiento del proceso de enseñanza-aprendizaje de Análisis Estructural a través de encuestas aplicadas a estudiantes en pruebas piloto.

## 2. Marco Pedagógico

### 2.1. Método Matricial de Rigidez

El método matricial de rigidez constituye la base del cálculo estructural computacional moderno, ya que permite formular de manera sistemática y general las ecuaciones de equilibrio que rigen el comportamiento de una estructura bajo carga mediante el álgebra matricial. Su potencia radica en su capacidad para automatizar el análisis de estructuras complejas, hiperestáticas y de gran escala, que serían imprácticas de resolver mediante métodos manuales (McGuire et al., 2015).

La ecuación fundamental del método es:

$$\{F\} = [k]\{U\} + \{F^{fem}\} \quad (1)$$

donde  $\{F\}$  es el vector de fuerzas nodales,  $[K]$  es la matriz de rigidez global de la estructura,  $\{U\}$  es el vector de desplazamientos nodales, y  $\{F^{fem}\}$  es el vector de fuerzas de empotramiento perfecto. El proceso involucra la discretización de la estructura, el ensamblaje de  $[K]$  a partir de las matrices de rigidez de cada elemento, la aplicación de condiciones de contorno, la reducción del sistema, la solución del sistema de ecuaciones y el cálculo de las fuerzas internas (Kassimali, 2015).

A diferencia del método de fuerzas (o método de flexibilidad), en el que las incógnitas primarias son las fuerzas redundantes del sistema, el método de rigideces toma como incógnitas los desplazamientos nodales. Esto implica que el proceso de ensamble de la matriz global, la imposición de condiciones de contorno y la resolución del sistema siguen un algoritmo uniforme, independientemente de la tipología estructural analizada (Derucher et al., 2021). Como característica del método, el tamaño del sistema de ecuaciones crece con el número de grados de libertad de la estructura, lo que en estructuras de gran escala puede representar un mayor costo

computacional frente al método de fuerzas cuando el grado de hiperestaticidad es bajo (Dumka et al., 2025; Mandal, 2026).

## **2.2. Lineamientos Curriculares para la Formación en Ingeniería Civil**

La formación de ingenieros en el contexto actual demanda una articulación entre fundamentos disciplinares y competencias tecnológicas. Los programas de ingeniería deben formar profesionales con capacidad científica y tecnológica, así como con habilidades para adaptar e implementar desarrollos computacionales que aporten a la solución de problemas del campo (Ahmed et al., 2021; Mendoza Carranza et al., 2024).

En el ámbito de la acreditación internacional, organismos como ABET establecen resultados de aprendizaje verificables que los programas de ingeniería deben garantizar. Entre estos se encuentran: la capacidad de identificar, formular y resolver problemas complejos de ingeniería; la capacidad de adquirir y aplicar nuevos conocimientos; y la capacidad de desarrollar experimentación, analizar e interpretar datos (ABET, 2021). Estos lineamientos orientan el diseño curricular hacia una formación integral que trasciende la mera transmisión de contenidos.

Siguiendo esta orientación hacia una formación integral, la exposición temprana a múltiples entornos computacionales favorece el desarrollo de competencias transferibles a otros ámbitos de la práctica profesional (Talaat et al., 2022). Esta formación no se limita al manejo operativo de software, sino que busca que los estudiantes comprendan los fundamentos que subyacen a las herramientas que utilizan.

## **2.3. Estrategias de Enseñanza con Herramientas Computacionales**

La enseñanza en programas de ingeniería enfrenta el desafío de articular fundamentos teóricos con herramientas computacionales. Las habilidades de programación en ingeniería civil deben diferenciarse de aquellas de otras disciplinas: su propósito no es el desarrollo de software

en sí mismo, sino la aplicación de principios de ingeniería para resolver problemas del campo mediante el uso computacional (Bowen, 2003).

La programación ofrece un espacio de aprendizaje propicio para que los estudiantes exploren e interactúen de forma profunda con los conceptos fundamentales de una disciplina. Esta práctica fomenta un pensamiento lógico y algorítmico para la resolución de problemas, incrementa la motivación y permite experimentar de manera versátil con los principios teóricos (Minett & Perera, 2006).

La elección del lenguaje de programación en contextos educativos no es neutra desde el punto de vista pedagógico. Python presenta ventajas significativas para la enseñanza en ingeniería: su sintaxis clara y legible reduce la curva de aprendizaje, su condición de código abierto elimina barreras de acceso, y su ecosistema de bibliotecas científicas permite abordar problemas de ingeniería con relativa facilidad (Chimarro-Amaguaña et al., 2023; Downey, 2024). Estas características lo hacen particularmente adecuado para entornos educativos, donde se busca que el estudiante se concentre en la aplicación de conceptos de ingeniería más que en las complejidades sintácticas del lenguaje.

#### **2.4. Metodologías**

El Aprendizaje Basado en Problemas (ABP) constituye una de las metodologías activas ampliamente adoptadas en la enseñanza de la ingeniería. Esta aproximación promueve que los estudiantes desarrollen conocimientos y habilidades a través de la resolución de problemas auténticos y contextualizados, en contraste con modelos tradicionales centrados en la transmisión unidireccional de contenidos. El ABP permite que los estudiantes enfrenten situaciones progresivamente complejas que demandan la integración de conceptos teóricos, habilidades prácticas y criterio profesional. Esta metodología favorece el pensamiento crítico, la capacidad

para resolver problemas complejos y la motivación hacia el aprendizaje autónomo (Mendoza Carranza et al., 2024; Razo Abundis et al., 2024).

Un aspecto central del ABP es la organización de los contenidos en torno a problemas de complejidad creciente. La progresión pedagógica facilita que los estudiantes construyan su comprensión de manera gradual, partiendo de configuraciones sencillas que permiten la familiarización con el entorno de trabajo, hasta llegar a problemas que incorporan condiciones más complejas. Esta estructura escalonada reduce la carga cognitiva inicial y permite que cada nuevo concepto se apoye en los previamente adquiridos, lo que resulta particularmente relevante cuando los estudiantes se enfrentan simultáneamente al aprendizaje de nuevos conceptos y nuevas herramientas (Mendoza Carranza et al., 2024).

Por otro lado, trabajo colaborativo en el aula se fundamenta en la premisa de que la interacción entre pares favorece procesos de aprendizaje más profundos y significativos. Esta metodología, que complementa el ABP, se caracteriza por crear espacios de discusión y construcción colectiva del conocimiento, donde los estudiantes asumen un rol activo en su propio proceso formativo. Esta interacción entre pares favorece la identificación y corrección de errores, la explicación de conceptos en un lenguaje compartido y el desarrollo de habilidades comunicativas, desarrollando competencias como la comunicación efectiva, la responsabilidad compartida y la motivación estudiantil, además de optimizar el rendimiento académico cuando se implementa con un diseño pedagógico adecuado (Cochachi Puray et al., 2026; Núñez et al., 2026).

## **2.5. Evaluación del Aprendizaje**

La evaluación del aprendizaje en ingeniería ha transitado desde modelos centrados exclusivamente en la calificación de productos finales hacia enfoques integrados que consideran el proceso, la retroalimentación continua y el desarrollo de competencias. Esta evolución responde

a la necesidad de formar profesionales capaces no solo de reproducir conocimientos, sino de aplicarlos en contextos complejos y de autorregular su propio aprendizaje (Shrinivasan, 2025).

La diversificación de instrumentos de evaluación permite capturar distintas dimensiones del aprendizaje. La combinación de evaluaciones de bajo riesgo aplicadas de manera frecuente, como cuestionarios cortos o ejercicios prácticos, favorece la retención de conocimientos, la motivación sostenida y la autorregulación del estudio (Gasull et al., 2025).

### 3. Diseño Metodológico del Proyecto de Aula

#### 3.1. Contenidos Desarrollados

Los contenidos desarrollados se organizaron en torno a nueve casos de estudio de sistemas en dos dimensiones, clasificados en tres tipologías estructurales (armaduras, pórticos y sistemas mixtos) con tres casos por cada una. Cada conjunto de tres casos se distribuyó a su vez en tres niveles de complejidad creciente: bajo, intermedio y avanzado, de manera que el total de nueve casos cubriera de forma progresiva el espectro de situaciones que, en el desarrollo habitual de la asignatura, pueden no alcanzarse a tratar completamente.

Los casos de nivel bajo se diseñaron para permitir la familiarización con el entorno de programación y el flujo de trabajo general del método matricial de rigidez incorporando configuraciones estructurales sencillas con cargas básicas. Los casos de nivel intermedio introdujeron condiciones adicionales como distintas idealizaciones de cargas sobre los elementos (cargas distribuidas abarcando varias tipologías, momentos y cargas puntuales), mayor cantidad de elementos y distintas configuraciones de restricciones, incrementando la complejidad del ensamble de la matriz global del sistema, el vector de fuerzas de empotramiento perfecto y la solución general del sistema. Los casos de nivel avanzado incorporaron elementos de mayor exigencia conceptual y computacional, propios de sistemas mixtos más cercanos a la realidad estructural. Entre estos elementos se incluyen: rótulas (articulaciones), que exigen modificaciones en la matriz de rigidez del elemento al liberar el grado de libertad rotacional; apoyos elásticos, modelados como resortes con rigidez asociada, los cuales buscan simular las relaciones de rigidez entre el suelo adyacente y la cimentación o el terreno de fundación; y, en el caso avanzado de sistemas mixtos, un elemento tipo resorte modelado como una barra con rigidez equivalente.

La selección de los nueve casos se realizó considerando tanto la progresión pedagógica como la representatividad de situaciones reales de ingeniería dentro del régimen elástico-lineal en dos dimensiones. Para verificar la coherencia de dimensiones, secciones y propiedades de cada caso, todos fueron modelados preliminarmente en SAP2000®, lo que permitió garantizar la validez de los datos de entrada antes de proceder con su implementación en Python y OpenSees.

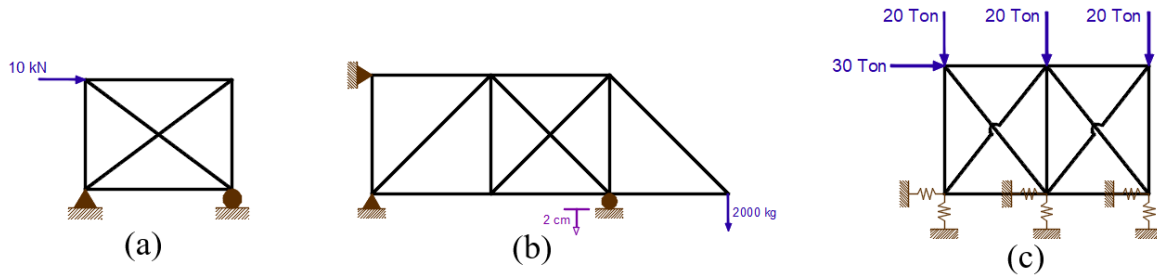
### ***3.1.1. Casos de estudio***

Con el fin de abarcar una amplia gama de fenómenos estructurales y permitir una progresión pedagógica clara, se definieron nueve casos de estudio, organizados en tres tipologías (armaduras, pórticos y sistemas mixtos) y tres niveles de complejidad (bajo, intermedio y avanzado). La selección de cada caso respondió a la inclusión de condiciones específicas que aumentan gradualmente la dificultad del análisis.

Los ejercicios definidos para armaduras (véase *Figura 1*) abordan condiciones de complejidad creciente. El caso de nivel bajo *Figura 1 (a)* corresponde a una armadura que, siendo externamente determinada, es internamente indeterminada, lo que permite al estudiante comprender que la hiperestaticidad no solo depende de los apoyos, sino también de la disposición interna de los elementos, introduciendo así el concepto de redundantes internas. El caso de nivel intermedio *Figura 1 (b)* incorpora un mayor número de elementos y condiciones de contorno más diversas, destacando la inclusión de un asentamiento diferencial en un apoyo, lo que obliga al estudiante a considerar los efectos de las deformaciones impuestas en el análisis más allá de las cargas externas. Y, el caso de nivel avanzado *Figura 1 (c)* introduce la combinación de múltiples cargas externas y la modelación de apoyos elásticos mediante resortes, desafiando al estudiante a comprender cómo la rigidez del apoyo influye en el ensamble de la matriz global.

**Figura 1**

*Casos de estudio definidos para armaduras*

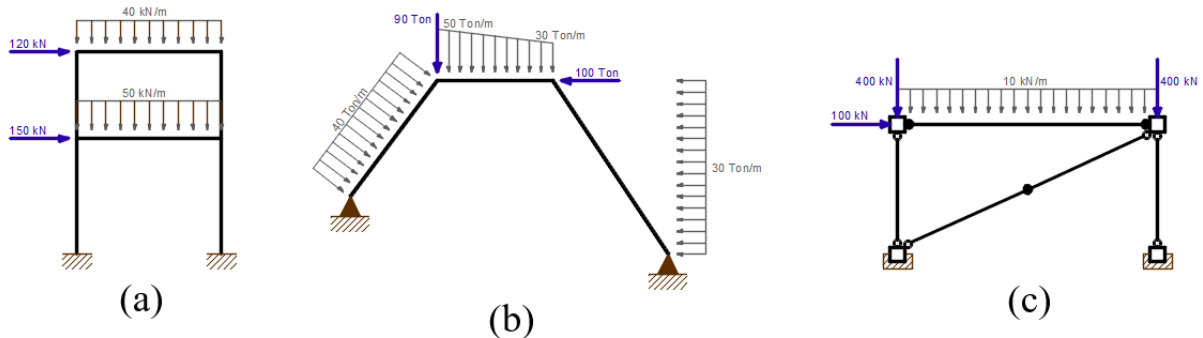


*Nota.* (a) Nivel bajo. (b) Nivel intermedio. (c) Nivel avanzado. Elaboración propia.

En cuanto a los pórticos (véase *Figura 2*), el caso de complejidad baja *Figura 2 (a)* presenta un pórtico de dos niveles con cargas distribuidas en vigas y fuerzas puntuales en los nodos, sirviendo como base para dominar el ensamblaje de matrices de rigidez en estructuras con tres grados de libertad por nodo con elementos verticales y horizontales. El caso de complejidad intermedia *Figura 2 (b)* aumenta la dificultad al incorporar elementos con distintas inclinaciones y condiciones de carga complejas; un desafío pedagógico clave en este ejercicio es la presencia de una carga distribuida que no está aplicada directamente sobre la proyección horizontal del elemento, lo que requiere un cuidado especial en el cálculo de las fuerzas de empotramiento perfecto, además de incluir condiciones de apoyo inusuales para pórticos, como apoyos fijos. El caso de complejidad avanzada *Figura 2 (c)* se centra en la modelación de conectividades especiales mediante rótulas o articulaciones en la unión de algunos elementos, donde el estudiante debe aprender a modificar la matriz de rigidez para liberar grados de libertad de rotación, un concepto fundamental para el análisis de estructuras más realistas.

**Figura 2**

*Casos de estudio definidos para pórticos*

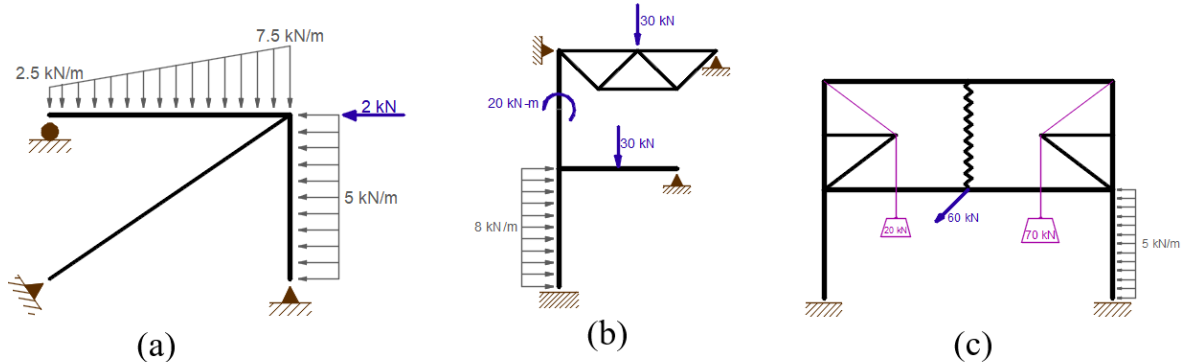


*Nota.* (a) Nivel bajo. (b) Nivel intermedio. (c) Nivel avanzado. Elaboración propia.

Para los sistemas mixtos (véase *Figura 3*), el caso de nivel bajo *Figura 3 (a)* corresponde a un sistema simple que combina una viga, una columna y una barra, con la particularidad de incluir un apoyo inclinado que introduce la necesidad de transformar coordenadas en las condiciones de contorno, un concepto que no aparece en los ejercicios anteriores. El caso de nivel intermedio *Figura 3 (b)* presenta una configuración donde la tipología de los elementos es claramente diferenciable entre vigas, columnas y barras, y se combinan múltiples tipos de carga sobre los elementos, incluyendo cargas distribuidas, fuerzas y momentos puntuales, lo que exige un correcto ensamblaje del vector de fuerzas de empotramiento perfecto. Y, el caso de nivel avanzado *Figura 3 (c)* resulta particularmente desafiante por la cantidad y diversidad de elementos que lo componen, incluyendo vigas, columnas, barras axiales y resortes, exigiendo que el estudiante demuestre su capacidad para integrar todos los conceptos aprendidos en un único modelo y resolver así un sistema estructural complejo y cercano a situaciones reales.

**Figura 3**

*Casos de estudio definidos para sistemas mixtos*



*Nota.* (a) Nivel bajo. (b) Nivel intermedio. (c) Nivel avanzado. Elaboración propia.

Si bien los nueve casos de estudio se plantearon como ejercicios académicos en el régimen elástico-lineal bidimensional, las condiciones estructurales que incorporan tienen una correspondencia directa con situaciones que se presentan en la práctica de la ingeniería civil.

Los apoyos articulados (rótulas) representan conexiones en las que se libera el grado de libertad rotacional, lo que en la realidad corresponde a uniones en las que la transmisión de momento es despreciable o indeseable. Ejemplos frecuentes son las conexiones entre vigas y columnas en estructuras metálicas con uniones atornilladas simples, y las articulaciones en puentes y arcos que se diseñan deliberadamente para permitir la rotación y reducir los momentos de empotramiento (He et al., 2023; Nawar et al., 2021).

Los apoyos con asentamiento diferencial modelan la situación en la que un apoyo experimenta un desplazamiento vertical impuesto, lo que ocurre cuando los suelos de fundación presentan heterogeneidades o cuando se producen fenómenos de consolidación diferencial. Esta condición es relevante en estructuras sobre suelos blandos o en edificaciones que experimentan asentamientos no uniformes en su cimentación (Chen et al., 2022).

Los apoyos elásticos tipo resorte aproximan la rigidez del suelo o del elemento de cimentación que soporta la estructura, en lugar de asumir una restricción rígida o completamente libre. Esta idealización es propia del enfoque de interacción suelo-estructura, en el que la cimentación se modela como un medio con rigidez, y es aplicable en el diseño de losas de cimentación, pilotes y estructuras sobre suelos con módulo de reacción definido (Elbalghiti et al., 2025; Mourya et al., 2023).

En cuanto a las condiciones de carga, las cargas distribuidas sobre elementos representan el peso propio de losas, cubiertas o muros transmitidos a vigas; las cargas puntuales en nodos corresponden a cargas concentradas de columnas, vigas secundarias o equipos; y los momentos en nodos pueden representar excentricidades de carga o condiciones de empotramiento parcial. Los sistemas mixtos, combinación de elementos tipo armadura y pórtico, tienen correspondencia con estructuras reales en las que coexisten componentes de diferente rigidez a la flexión: por ejemplo, pórticos de concreto reforzado con arriostramientos metálicos, o estructuras industriales que combinan vigas-columna con cables o barras de arriostramiento.

### ***3.1.2. Resultados de la validación computacional***

La validación de los algoritmos implementados en Python se realizó mediante comparación cruzada con los modelos desarrollados en OpenSees y con los resultados de referencia obtenidos en SAP2000®, siguiendo el criterio de aceptación de un margen de error máximo del 5% establecido para el proyecto. Los resultados detallados de cada caso se presentan en el Apéndice B; a continuación, se sintetizan los hallazgos principales por tipología estructural.

En los casos de armaduras (véase Tablas B1 a B3), la comparación entre OpenSees y SAP2000® arrojó un error máximo del 0.15%, correspondiente al tercer ejercicio, que representa el caso de mayor complejidad de esta tipología, posiblemente debido al tratamiento que les da cada

software a los apoyos elásticos, ya que en el caso de OpenSees se modelaron como elementos tipo “zeroLength”. La comparación entre el método matricial implementado en Python y OpenSees mostró errores del 0.00% en los tres casos, lo que evidenció una correspondencia exacta entre ambos entornos de cálculo y confirmó la correcta implementación del algoritmo matricial para esta tipología.

En los casos de pórticos (véase Tablas B4 a B6), el error máximo entre OpenSees y SAP2000® fue del 1.65%, igualmente en el tercer ejercicio, en SAP2000® se forzó a que el momento en los elementos donde se presentaban rótulas fuera cero, mientras que en OpenSees se modeló como elementos “zeroLength” cuya rigidez correspondiente a la rotación era cercana a cero, si se usaba cero la matriz sería singular y el sistema de ecuaciones no podría resolverse. La comparación entre Python y OpenSees mostró un error del 0.00% en los tres casos, lo que evidencia una coincidencia total en los resultados obtenidos con ambas herramientas.

En los casos de sistemas mixtos (véase Tablas B7 a B9) se registraron los mayores porcentajes de error de todo el estudio, con un valor máximo del 2.32% entre OpenSees y SAP2000®, en el segundo ejercicio. La comparación entre Python y OpenSees se mantuvo en 0.00% para los tres casos de esta tipología, lo que indica que los algoritmos matriciales implementados reproducen fielmente el comportamiento calculado por OpenSees. La mayor discrepancia respecto a SAP2000® en esta tipología es consistente con la mayor complejidad del ensamblaje matricial que caracteriza a los sistemas mixtos, en los que coexisten elementos tipo armadura y elementos tipo pórtico con diferentes formulaciones de rigidez. No se descarta, que diferencias en el tratamiento interno de las condiciones de carga entre los dos softwares hayan contribuido a las discrepancias observadas, aunque este aspecto excede los alcances del presente proyecto.

En conjunto, los nueve casos de estudio se mantuvieron dentro del margen de error del 5% en todas las comparaciones realizadas, con un error máximo global del 2.32% entre OpenSees y SAP2000®, y del 0.00% entre el método matricial en Python y OpenSees. Estos resultados respaldan la confiabilidad de los algoritmos desarrollados y validan su uso como material docente para la enseñanza del análisis estructural, garantizando que los procedimientos implementados son técnicamente correctos y consistentes con herramientas de referencia.

### 3.2. Material Didáctico

El material didáctico se organizó en torno a los dos ejes computacionales del proyecto: la implementación del método matricial en Python y su modelo de validación en OpenSeesPy. Para cada caso de estudio se elaboraron, en relación con estos dos códigos, guías explicativas en formato PDF, diapositivas y cuestionarios de verificación del aprendizaje. Completaron el material una serie de videos tutoriales centrados en los casos más representativos de cada tipología estructural y una guía de instalación de Anaconda® (véase Apéndice C).

Los códigos desarrollados en Python para el método matricial de rigidez siguieron una estructura lineal, con variables de nombres autoexplicativos y secciones claramente delimitadas, privilegiando un enfoque pedagógico por encima de la eficiencia computacional. Cada código incluyó la definición de datos de entrada (propiedades de materiales y secciones, coordenadas de nodos, conectividad de elementos y grados de libertad del sistema), el ensamble de la matriz de rigidez global, la aplicación de condiciones de contorno, la reducción del sistema, la solución del sistema de ecuaciones, la presentación organizada de resultados en consola y la visualización gráfica del sistema deformado. Para los casos de pórticos y sistemas mixtos se incorporó adicionalmente el cálculo del vector de fuerzas de empotramiento perfecto y los diagramas de fuerzas internas. Los modelos en *openseespy* siguieron un flujo equivalente, empleando las

funciones propias del framework para definir nodos, elementos, materiales, condiciones de apoyo y cargas, y extrayendo los mismos resultados para su comparación con los obtenidos en Python.

Las guías explicativas en formato PDF (véase Apéndice D) presentaron el código fuente completo acompañado de comentarios que explicaron cada sección, permitiendo al estudiante seguir la lógica del programa sin necesidad de ejecutarlo. Las diapositivas (véase Apéndice E) desarrollaron el proceso de análisis desde una perspectiva conceptual, abordando la discretización del ejercicio e identificación de información relevante, sirvieron como guion y apoyo visual para la producción de los videos. Los videos tutoriales (véase Apéndice F) explicaron el flujo general del programa y el significado de las variables de manera dinámica, sin limitarse a leer los comentarios línea por línea.

Todo el material fue organizado y puesto a disposición de los estudiantes a través de una página estructurada en el aula virtual de aprendizaje de Moodle (véase Apéndice G). Los videos tutoriales fueron alojados en YouTube y OneDrive con acceso restringido mediante enlace directo, de manera que el material permanece disponible únicamente para la directora del proyecto, la biblioteca de la universidad, los estudiantes de la asignatura de Análisis Estructural y la autora, dado su carácter estrictamente interno y académico.

En cuanto al uso autónomo del material, el tiempo que cada estudiante requiera para trabajarlo depende de variables que no es posible prever ni estandarizar: la experiencia previa en programación, las dificultades de instalación y configuración del entorno, el tiempo necesario para comprender el flujo del programa y la capacidad de interpretar y adaptar los códigos a nuevas situaciones. Más que una estimación de horas, lo que determina la apropiación del material es la dedicación sostenida que el estudiante pueda comprometer con su propio proceso de aprendizaje. En cuanto a las formas de evaluación recomendadas en la literatura para este tipo de material, la

combinación de evaluaciones frecuentes de bajo riesgo, como cuestionarios cortos, ejercicios de modificación de código o preguntas de interpretación de resultados, favorece la retención de conocimientos y la autorregulación del estudio (Gasull et al., 2025), aproximación coherente con los cuestionarios de verificación incluidos en cada caso de estudio del presente proyecto.

### **3.3. Estrategias de Evaluación**

La evaluación del aprendizaje se diseñó a partir de una estrategia mixta que combinó instrumentos cuantitativos y cualitativos, aplicados en distintos momentos del proceso con el fin de obtener una visión integral de la pertinencia y utilidad del material docente desarrollado.

Como punto de partida, se aplicó una encuesta diagnóstica a 55 estudiantes de la asignatura Análisis Estructural en el periodo 2025-2, mediante un cuestionario estructurado con preguntas cerradas sobre conocimientos previos en programación, frecuencia de uso de herramientas computacionales y percepción sobre su utilidad en el análisis estructural (véase Apéndice A). Este instrumento permitió caracterizar el punto de partida del grupo y orientar decisiones de diseño del material.

Para la prueba piloto, se aplicó previamente una encuesta de inscripción a los interesados en participar con solicitud de divulgación en los diferentes canales de la escuela, mediante la cual se recolectaron datos sobre el semestre actual, la experiencia previa con Python y OpenSees, y con qué docente habían cursado la asignatura de Análisis Estructural (véase Apéndice H). Esta información permitió conformar un grupo focal con criterios de inclusión definidos y homogéneos.

Durante el desarrollo de las sesiones, se emplearon encuestas de seguimiento aplicadas al inicio de cada encuentro, en las que los participantes calificaron la sesión anterior, expresaron su percepción sobre los temas tratados en términos de ritmo y dificultad, y dejaron comentarios y sugerencias en un espacio abierto (véase Apéndice I). Este instrumento cumplió también la función

de registro de asistencia y constituyó la principal fuente de retroalimentación continua para el ajuste progresivo del material. Al finalizar la prueba piloto, se aplicó una encuesta de evaluación final orientada a valorar globalmente la experiencia (véase Apéndice J).

La observación directa durante las diez sesiones presenciales constituyó un instrumento cualitativo fundamental. A través de ella se identificaron dudas recurrentes, patrones de dificultad y niveles de interacción con el material, información que fue registrada y analizada para orientar los ajustes realizados entre sesiones. Las sesiones fueron grabadas mediante captura de pantalla, con el doble propósito de constituir un recurso adicional y facilitar el análisis posterior de las dinámicas de aprendizaje observadas.

Todos los participantes fueron notificados sobre los objetivos del proyecto y firmaron un consentimiento informado (véase Apéndice K) en el que autorizaron la grabación de las sesiones y el uso de sus respuestas con fines académicos. Se garantizó en todo momento la confidencialidad de la información y la participación voluntaria, conforme a los principios éticos que rigen la investigación con seres humanos establecidos por el Comité de Ética en Investigación Científica (CEINCI) de la Universidad Industrial de Santander (Solano et al., 2010).

## 4. Desarrollo de la Experiencia Docente

### 4.1. Actividades Realizadas

Para conformar el grupo participante en la prueba piloto se aplicó una encuesta de inscripción (véase Apéndice H), a través de la cual se preinscribieron 52 estudiantes. A partir de ese listado se realizó un filtrado priorizando a quienes ya habían cursado la asignatura de Análisis Estructural, y teniendo en cuenta el límite de aforo del aula disponible.

La prueba piloto tuvo una duración de diez sesiones de aproximadamente dos horas cada una. La primera sesión estuvo enfocada en la contextualización del proyecto, la presentación del flujo general para la resolución de los ejercicios y, la instalación y configuración del entorno de trabajo mediante Anaconda, incluyendo Jupyter Notebook y las librerías requeridas. Dado que no todos los participantes estuvieron presentes en esta sesión y algunos no pudieron completar la instalación por las características de sus equipos, se les proporcionó la guía de instalación para que realizaran el proceso de forma independiente.

A partir de la segunda sesión se guió la solución del ejercicio 3 de armaduras (nivel avanzado) mediante el método matricial de rigidez, empleando tres sesiones para su desarrollo completo en Python. Las sesiones 5 y 6 estuvieron dedicadas a la construcción y análisis del modelo equivalente en *openseespy*, permitiendo la validación cruzada de los resultados obtenidos. Durante este proceso se identificó que las características del ejercicio, con apoyos elásticos en lugar de apoyos convencionales, impedían ilustrar el proceso de reducción del sistema de ecuaciones, uno de los procedimientos fundamentales del método matricial, lo que motivó la decisión de abordar el ejercicio de nivel básico de pórticos en lugar del caso de mayor complejidad inicialmente previsto.

En las sesiones restantes se guió la solución del ejercicio 1 de pórticos (nivel básico) mediante el método matricial de rigidez, cubriendo desde la definición de datos de entrada hasta la generación de los diagramas de fuerza axial, cortante y momento flector.

A lo largo de todas las sesiones se aplicó al inicio una encuesta de seguimiento en la que los participantes calificaron el encuentro anterior y dejaron comentarios y sugerencias (véase Apéndice I).

#### **4.2. Interacción con Estudiantes**

El desarrollo de las sesiones se dio en torno a un diálogo cordial y respetuoso, en el que se guiaba la construcción del código y los participantes podían manifestar en cualquier momento sus dudas o los errores que no lograban resolver de forma independiente. La dinámica del grupo fue variable: hubo sesiones con alta participación e intercambio fluido de preguntas, y otras en las que el grupo se mantuvo más pasivo, siguiendo el desarrollo del código sin intervenir con frecuencia.

Las dudas más recurrentes estuvieron relacionadas con aspectos de sintaxis y errores en Python, así como con dificultades técnicas asociadas a la configuración de los equipos y el entorno de trabajo. Estas situaciones fueron atendidas en el momento, ajustando el ritmo de la explicación según las necesidades del grupo. En un caso particular, un participante no pudo completar la instalación de Anaconda debido a complicaciones con el equipo, por lo que se le brindó acompañamiento personalizado en un horario fuera de las sesiones regulares.

A medida que avanzaron las sesiones y la asistencia fue disminuyendo, se adoptaron estrategias para facilitar la continuidad del proceso: al inicio de cada encuentro se compartía el código trabajado en la sesión anterior y se realizaba un breve resumen del avance acumulado, permitiendo que quienes no habían asistido pudieran retomar el hilo sin perder el contexto del ejercicio.

La retroalimentación recogida a través de las encuestas de seguimiento complementó la interacción presencial, permitiendo identificar entre sesiones aspectos que no siempre emergían durante las clases, como la necesidad de profundizar en la explicación de comandos y funciones específicas de Python y OpenSeesPy.

### **4.3. Evaluación del Proceso**

#### ***4.3.1 Asistencia***

El resumen de asistencia a lo largo de las diez sesiones se presenta en el Apéndice I. La participación fue variable durante el proceso: la sesión 3 registró la asistencia más alta con el 83.9% de los 31 inscritos, mientras que las sesiones finales presentaron una asistencia entre el 16.1% y el 22.6%, con excepción de la sesión 9 que registró un repunte del 29.0% (Tabla I1).

La irregularidad en la asistencia tuvo implicaciones directas sobre la experiencia de aprendizaje. Los participantes que faltaban a una o varias sesiones llegaban a los encuentros siguientes sin el contexto acumulado del ejercicio en desarrollo, lo que dificultaba su seguimiento y comprensión, especialmente en un proceso de programación secuencial donde cada sesión construye sobre la anterior. Esta situación se refleja en las valoraciones del cuestionario final: los participantes con menor asistencia tendieron a calificar el material y la experiencia en las categorías de deficiente o muy deficiente, mientras que quienes asistieron con mayor regularidad otorgaron valoraciones predominantemente positivas. En este sentido, las calificaciones más bajas no responden necesariamente a deficiencias en el material o en la dinámica de las sesiones, sino que están directamente relacionadas con una participación insuficiente para apropiarse del conocimiento de forma progresiva.

#### ***4.3.2 Seguimiento semanal***

Las encuestas de seguimiento (Apéndice I) fueron aplicadas al inicio de cada sesión a partir de la tercera. Cada encuesta incluyó una calificación general de la sesión anterior en escala de 1 a 5, una valoración de la dificultad percibida de los temas tratados y un espacio abierto para comentarios y sugerencias.

Los resultados evidenciaron una percepción positiva y sostenida a lo largo de todo el proceso. En una escala de 1 a 5, donde 1 es la valoración más baja y 5 la más alta, la calificación promedio por sesión osciló entre 4.60 y 5.00, sin registrar descensos significativos en las etapas de mayor complejidad temática (Tabla I2). La sesión 9 fue la mejor valorada con una calificación promedio de 5.00, mientras que la sesión 8 obtuvo la calificación más baja con 4.60, valor que sigue siendo considerablemente alto dentro de la escala empleada. En cuanto a la dificultad percibida, la gran mayoría de los participantes calificó los temas como adecuados en todas las sesiones, con casos aislados que los señalaron como difíciles únicamente en las sesiones 4 y 9 (Tabla I3), lo que sugiere que la progresión pedagógica diseñada fue apropiada para el nivel de los participantes.

Los comentarios abiertos permitieron identificar patrones recurrentes que orientaron ajustes progresivos al material entre sesiones. La tendencia más frecuente fue la ausencia de comentarios o sugerencias específicas, interpretada como una señal de conformidad general con el desarrollo de los encuentros. Entre los comentarios sustantivos, la observación más reiterada fue la necesidad de profundizar en la explicación de los comandos y funciones de Python y OpenSeesPy, indicando que no era suficiente con nombrar la librería o función empleada, sino que se requería explicar con mayor detalle qué hacía cada instrucción dentro del contexto del ejercicio.

Esta retroalimentación motivó ajustes progresivos al material, ampliando los comentarios en los códigos e incorporando explicaciones más detalladas en las guías PDF.

#### ***4.3.3 Evaluación del Material***

El cuestionario final, cuyo instrumento completo se presenta en el Apéndice J, fue respondido por 14 de los 31 participantes inscritos. De estos, 6 reportaron haber asistido entre 1 y 4 sesiones, y 8 entre 5 y 8 sesiones (Tabla J1). En cuanto a la experiencia previa en programación, 5 participantes no tenían ninguna experiencia, 8 contaban con experiencia básica y 1 con experiencia intermedia (Tabla J2).

La evaluación del material docente arrojó resultados favorables en todos los ítems valorados (Tabla J3). Las guías explicativas en PDF y las presentaciones didácticas fueron los recursos mejor calificados, con 9 respuestas en la categoría de excelente en cada caso. Los códigos en Python y los modelos en *openseespy* fueron valorados positivamente por la mayoría, aunque concentraron un mayor número de respuestas en las categorías de deficiente y regular en comparación con los demás recursos, lo que es coherente con las dificultades reportadas en torno a la comprensión de la sintaxis y los comandos empleados, y con el hecho de que varios respondientes tuvieron una asistencia limitada que dificultó su apropiación de estos elementos. La dinámica y el ritmo de las sesiones presenciales fue el ítem con mayor dispersión en las respuestas, reflejando la heterogeneidad del grupo en términos de experiencia previa en programación: quienes contaban con alguna familiaridad con Python tendieron a valorarla positivamente, mientras que quienes no tenían experiencia previa encontraron el ritmo más exigente.

#### ***4.3.4 Evaluación de la Experiencia Docente***

La disponibilidad del auxiliar docente para resolver dudas, el ambiente de confianza y participación, y la utilidad de la retroalimentación recibida fueron los aspectos mejor valorados de

la experiencia docente, con 11 respuestas de excelente en cada uno de estos ítems (véase Apéndice J). La claridad de las explicaciones durante las sesiones obtuvo una distribución más amplia, con respuestas concentradas principalmente en bueno y excelente, pero con casos en regular y muy deficiente que coinciden con las dificultades señaladas en torno al ritmo y la profundidad de las explicaciones de los comandos de Python (Tabla J4).

En este punto es pertinente retomar el análisis de la asistencia: los participantes que calificaron la experiencia docente en las categorías más bajas corresponden en su mayoría a quienes asistieron a un número reducido de sesiones. La naturaleza secuencial y acumulativa del contenido implicaba que cada sesión dependía de la comprensión de las anteriores, por lo que la asistencia irregular generó brechas de comprensión que naturalmente incidieron en la percepción de la claridad de las explicaciones. Esta consideración no invalida las observaciones recibidas, sino que contextualiza su origen y orienta mejoras específicas para futuras implementaciones, particularmente en lo que respecta a la introducción progresiva a la sintaxis de Python antes de abordar los ejercicios de análisis estructural.

#### ***4.3.5. Aprendizaje Percibido***

Los resultados del aprendizaje percibido reflejan una valoración positiva de la experiencia en términos de utilidad y proyección (véase Apéndice J). El 85.7% de los encuestados estuvo de acuerdo o totalmente de acuerdo en que lo aprendido en Python les será útil en otras asignaturas o en su futuro profesional, y el 92.9 % recomendaría incorporar el material en el curso de Análisis Estructural, lo que constituye un respaldo significativo a la pertinencia de la iniciativa (Tabla J5).

Respecto a la autonomía en la implementación del método matricial en Python, las respuestas reflejaron una apropiación parcial, consistente con el nivel introductorio de la experiencia. La mayoría de los participantes indicó que podría avanzar apoyándose en los códigos

y guías desarrollados, pero reconoció limitaciones para implementar el método desde cero sin material de referencia. Este resultado es esperable dado que los participantes enfrentaron simultáneamente dos curvas de aprendizaje: la del lenguaje de programación y la implementación computacional del método matricial, lo que hace razonable que la autonomía plena requiera práctica adicional más allá del alcance de la prueba piloto.

#### ***4.3.6. Comentarios Cualitativos***

Los comentarios cualitativos del cuestionario final (véase Apéndice J) aportaron información valiosa para interpretar los resultados cuantitativos y orientar mejoras al material. En cuanto al recurso más valioso, las sesiones presenciales fueron destacadas por la mayoría de los participantes como el elemento central de la experiencia, por la posibilidad de resolver dudas en tiempo real y seguir el proceso paso a paso con acompañamiento directo. Las guías en PDF fueron valoradas como complemento fundamental para quienes no podían tomar notas con suficiente detalle durante las sesiones, y los videos fueron señalados como un recurso que facilita el repaso autónomo del contenido.

Entre las sugerencias de mejora predominó de forma consistente la necesidad de dedicar tiempo explícito a la introducción a la sintaxis de Python antes de abordar los ejercicios, explicando el propósito de cada comando y librería empleada en lugar de asumir familiaridad previa con el lenguaje. Algunos participantes señalaron que en ocasiones los códigos compartidos en las guías no coincidían exactamente con los desarrollados en sesión, lo que dificultaba retomar el proceso cuando se había perdido parte de una clase. Si bien durante las sesiones el auxiliar docente siguió las guías como referencia, es posible que pausas explicativas, variaciones menores en el orden de exposición o diferencias en la interpretación del participante hayan generado esta percepción. En cualquier caso, este señalamiento constituye una oportunidad de mejora para futuras

implementaciones, en las que podría considerarse compartir el código trabajado en sesión al finalizar cada encuentro, de forma que los participantes cuenten con una referencia exacta de lo desarrollado.

Varios participantes destacaron en sus comentarios adicionales el valor de incorporar herramientas computacionales en la formación del ingeniero civil, resaltando que el dominio de la programación representa una habilidad diferenciadora en el ámbito profesional. Estos comentarios refuerzan la pertinencia del proyecto más allá de los resultados cuantitativos obtenidos y constituyen un argumento sólido para su implementación formal en el curso de Análisis Estructural.

#### **4.4. Evidencias de Aprendizaje**

Las evidencias del proceso de aprendizaje desarrollado durante la prueba piloto se encuentran distribuidas a lo largo del documento y sus apéndices. Los resultados de las encuestas de seguimiento y del cuestionario final, analizados en la sección 4.3, constituyen la evidencia cuantitativa y cualitativa más directa de la experiencia.

El registro fotográfico del proceso se ilustra en la *Figura 4*, que registra el desarrollo de algunas sesiones presenciales. Las grabaciones editadas de las sesiones, que forman parte del material docente desarrollado, están disponibles como recurso adicional en el Apéndice F.

**Figura 4**

*Desarrollo de sesiones presenciales de la prueba piloto.*



### 5. Reflexión Pedagógica

La experiencia vivida a lo largo de esta práctica docente transformó de manera significativa la comprensión que se tenía sobre el acto de enseñar. Antes de iniciarla, la enseñanza se concebía fundamentalmente como una transferencia de conocimiento: quien sabe, explica; quien no sabe, aprende. Sin embargo, el contacto real con el aula evidenció que enseñar es un proceso de construcción mutua, en el que quien enseña también aprende, se cuestiona y se transforma. Cada sesión exigió no solo preparación técnica, sino sensibilidad para leer al grupo, identificar dónde estaban las dificultades reales y ajustar el discurso en consecuencia. En ese sentido, el dominio técnico del método matricial y de las herramientas computacionales, aunque necesario, resultó insuficiente por sí solo: la práctica docente demostró que saber no equivale automáticamente a saber enseñar.

El proceso fue demandante y en ocasiones laborioso, pues implicó sostener simultáneamente la preparación del material, la conducción de las sesiones y la gestión de la inseguridad frente al grupo. No obstante, resultó igualmente gratificante: ver a los estudiantes comprender un concepto que minutos antes les generaba confusión, o recibir una pregunta que evidenciaba un nivel de reflexión profundo sobre el contenido, constituyeron momentos de genuina satisfacción. La práctica docente abrió además una puerta inesperada: la posibilidad de desempeñar un rol docente en el futuro, algo que no estaba en el horizonte al iniciarla.

En cuanto al material desarrollado, los resultados fueron parciales. Los recursos demostraron ser útiles y bien valorados por los participantes que asistieron regularmente a las sesiones, y la progresión pedagógica por niveles de complejidad funcionó según lo esperado. Sin embargo, la evaluación final evidenció una brecha significativa en la percepción del material: quienes asistieron a un número reducido de sesiones tendieron a calificarlo en las categorías más

bajas, mientras que quienes tuvieron una participación constante otorgaron valoraciones predominantemente positivas. Esta situación refleja que la apropiación del conocimiento en un proceso secuencial y acumulativo como la programación requiere continuidad, y que la asistencia irregular constituye una limitación difícil de subsanar mediante el diseño instruccional.

Quedó pendiente una aspiración más profunda: que el estudiante no se limitara a comprender lo explicado, sino que se apropiara de la programación como herramienta propia y la explorara más allá de los casos propuestos, siguiendo sus propios intereses. Lograr ese nivel de motivación intrínseca en un grupo heterogéneo y en un tiempo limitado resultó más difícil de lo anticipado, y constituye un horizonte de mejora para futuras versiones del material.

Más allá de los resultados concretos, esta práctica reafirmó una convicción: integrar Python en la enseñanza del análisis estructural no es simplemente una decisión técnica, sino una apuesta pedagógica con implicaciones profundas. Cuando un estudiante programa su propio algoritmo, deja de ser usuario pasivo de un software y se convierte en autor de su propio proceso de cálculo, comprendiendo cada paso desde adentro. En un contexto profesional donde la velocidad, la eficiencia y la automatización son exigencias crecientes, dotar al futuro ingeniero civil de competencias en programación no es un lujo ni un complemento opcional: es prepararle para un mundo que ya llegó. Y si esa preparación logra además despertar en él la curiosidad por seguir explorando, por preguntarse qué más puede hacer con una línea de código, entonces el objetivo pedagógico más importante habrá sido cumplido.

**Referencias Bibliográficas**

- ABET*. (2021). Accreditation Board of Engineering and Technology. <https://www.abet.org/>
- Ahmed, D., Nayeemuddin, M., Ayadat, T., & Asiz, A. (2021). Computing Competency for Civil Engineering Graduates: Recent Updates and Developments in Saudi Arabia and the US. *International Journal of Higher Education*, 10(6), 57. <https://doi.org/10.5430/ijhe.v10n6p57>
- Chen, Q., Yang, Z., Xue, B., Wang, Z., & Xie, D. (2022). Monitoring and Prediction Analysis of Settlement for the Substation on Soft Clay Foundation. *Advances in Civil Engineering*, 2022, 1-15. <https://doi.org/10.1155/2022/1350443>
- Chimarro-Amaguaña, J. D., Chuqui-Barriga, F. A., Guamán-Cullispuma, D. P., & Quishpe-Farinango, C. I. (2023). El auge exponencial del lenguaje Python en el desarrollo tecnológico. *Revista Científica INGENIAR: Ingeniería, Tecnología e Investigación*. ISSN: 2737-6249., 6(12), 240-256.
- Cochachi Puray, J. N., Andia Cochachi, H. R., & Cañari Marticorena, H. F. (2026). Aprendizaje colaborativo en la educación superior: Una revisión sistemática sobre fundamentos, prácticas y desafíos actuales (2021–2025). *Revista InveCom*, 6(4), 4.
- Derucher, K., Putcha, C., Kim, U., & Hota, G. (2021). *Static Analysis of Determinate and Indeterminate Structures*. <https://doi.org/10.1201/9781003246633>
- Downey, A. B. (2024). *Think Python: How to Think like a Computer Scientist* (Third). O'Reilly Media. [https://alldowney-github-io.translate.google.com/ThinkPython/?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=es&\\_x\\_tr\\_hl=es&\\_x\\_tr\\_pto=tc](https://alldowney-github-io.translate.google.com/ThinkPython/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)
- Dumka, P., Mishra, D. R., Chauhan, R., & Pandey, N. (2025). Python-Powered Structural Analysis: Modeling and Solving 2D Truss Systems With the “Anastruct” Module.

*Computer Applications in Engineering Education*, 33(5), e70072.  
<https://doi.org/10.1002/cae.70072>

Elbalghiti, Y., Mkhaleet, M. E., & Lamdouar, N. (2025). Numerical Modeling of Soil-Structure Interaction: Case of Raft Foundations. *Civil Engineering and Architecture*, 13(1), 678-690.  
<https://doi.org/10.13189/cea.2025.130143>

Gasull, V. L., Savini, C., & Gimeno, P. B. (2025). Estudio de la implementación de metodologías activas y estrategias de evaluación en carreras de ingeniería. *Encuentro Internacional de Educación en Ingeniería*. <https://doi.org/10.26507/paper.4672>

He, J. C. W., Clifton, G. C., Ramhormozian, S., & Hogan, L. S. (2023). Numerical and analytical study of pinned column base plate connections. *Journal of Constructional Steel Research*, 204, 107846. <https://doi.org/10.1016/j.jcsr.2023.107846>

Kassimali, A. (2015). *Análisis estructural* (Quinta). Cengage Learning. <https://www-ebooks7-24-com.bibliotecavirtual.uis.edu.co/?il=1298>

Mandal, D. (2026). Development of an Interactive Web-Based Tool for 2D Truss Analysis Using the Direct Stiffness Method. *Computer Applications in Engineering Education*, 34.  
<https://doi.org/10.1002/cae.70183>

McGuire, W., Gallagher, R. H., & Ziemian, R. D. (2015). *Matrix Structural Analysis*. Wiley.  
<https://digitalcommons.bucknell.edu/cgi/viewcontent.cgi?article=1006&context=books>

McKenna, F., Fenves, G. L., Filippou, F. C., Mazzoni, S., Scott, M., Elgamal, A., Yang, Z., Lu, J., Arduino, P., McKenzie, P., Deierlein, G. G., & Law, K. (2025). *Open System for Earthquake Engineering Simulation*. <https://opensees.berkeley.edu/index.php>

- Mendoza Carranza, J. D., Mendoza Espinoza, S. A., & Romero Hidrovo, B. (2024). Planteo de una metodología para la enseñanza de las ciencias estructurales en ingeniería civil. *ResearchGate*. <https://doi.org/10.26507/rei.v19n38.1296>
- Minett, M. W., & Perera, C. (2006). *PROGRAMMING FOR CIVIL AND BUILDING ENGINEERS USING MATLAB*.
- Mourya, V. K., Pandey, G., Patel, D., & Kumar, R. (2023). Approaches considering non-linearity in soil-foundation-interaction: A State-of-the-Art Review. *Research on Engineering Structures & Materials*, 9(3), 989-1013.
- Nawar, M., Matar, E., Maaly, H., Alaaser, A., & Elzohairy, A. (2021). Assessment of Rotational Stiffness for Metallic Hinged Base Plates under Axial Loads and Moments. *Buildings*, 11, 368. <https://doi.org/10.3390/buildings11080368>
- Núñez, E. A. Á., Chávez, M. L. V., Corzo, C. R. A., & Mendoza, J. C. (2026). Estrategias y resultados del aprendizaje colaborativo en entornos universitarios. *Revista Tribunal*, 6(14), 168-187. <https://doi.org/10.59659/revistatribunal.v6i14.321>
- Razo Abundis, Y., Dibut Toledo, L., & Portal Gallardo, J. A. (2024). Las metodologías activas de aprendizaje en las carreras de Ingeniería. *Revista UGC*, 2(2), 46-51.
- Shrinivasan, S. (2025). A case study on redesigning classroom activities and assessments for Generation Z students. *Theses and Dissertations*. <https://scholarjunction.msstate.edu/td/6839>
- Solano, E. C., Alvarado, M. B. C., López, M. T. E., & Monroy, A. D. G. (2010). *Regulación ética en investigación con seres humanos en Colombia*. Comité de Ética en Investigación Científica (CEINCI), Universidad Industrial de Santander.

Talaat, A., Kohail, M., & Ahmed, S. (2022). *Programming in The Context of Civil Engineering Education*. <https://doi.org/10.21203/rs.3.rs-1802246/v1>

### **Apéndices**

Los apéndices están adjuntos y puede visualizarlos en la base de datos de la biblioteca UIS.